Claims

Having thus described our invention, what we claim as new and desire to secure by Letters Patent is as follows:

1. A method for processing network packets at an intermediate node, the method comprising:

forming a first connection between a first node and the intermediate node this first connection including:

a first flow originating at a first source flow end point on the first node and terminating at a first destination flow end point on the intermediate node, wherein processing at the first node associated with the first source flow end point conforms to a first protocol, and

a second flow originating at a second source flow end point on the intermediate node and terminating at a second destination flow end point on the first node, wherein processing at the first node associated with the second destination flow end point conforms to a second protocol;

forming a second connection between a second node and the intermediate node this second connection including:

a third flow originating at a third source flow end point on the intermediate node and

terminating at a third destination flow end point on the second node, wherein processing at the second node associated with the third destination flow end point conforms to a third protocol, and

a fourth flow originating at a fourth source flow end point on the second node and terminating at a fourth destination flow end point on the intermediate node, wherein processing at the second node associated with the fourth source flow end point conforms to a fourth protocol,

such that a given flow originates at a source flow end point on a source node and terminates at a destination flow end point on a destination node and data written to the source flow end point of a given flow can subsequently be read from the destination flow end point of the given flow without traversing any intervening flow end points, and

splicing the first flow and third flow to form a first composite flow originating at the first source flow end point on the first node and terminating at the third destination flow end point on the second node in a manner whereby the second flow and the fourth flow remain unchanged.

2. A method as recited in claim 1, wherein the step of splicing the first flow and third flow to form the first composite flow allows

processing at the first node associated with the first source flow end point to remain unmodified and continue to conform to the first protocol,

processing at the first node associated with the second destination flow end point to remain unmodified and continue to conform to the second protocol,

processing at the second node associated with the third destination flow end point to remain unmodified and continue to conform to the third protocol, and

processing at the second node associated with the fourth source flow end point to remain unmodified and continue to conform to the fourth protocol.

3. A method recited in claim 1, wherein the first node and the second node are the same node.

4. A method recited in claim 1, wherein the first node and intermediate node are the same node.

5. A method recited in claim 1, wherein the second node and the intermediate node are the same node.

6. A method recited in claim 1, wherein the first node, the second node, and the intermediate node are all the same node.

7. A method recited in claim 1, wherein the first protocol and the second protocol are the same protocol.

8. A method recited in claim 1, wherein the third protocol and the fourth protocol are the same protocol.

9. A method recited in claim 1, wherein the first protocol, the second protocol, the third protocol, and the fourth protocol are all the same protocol.

10. A method recited in claim 1, wherein the step of splicing the first flow and third flow to form the first composite flow includes:

identifying a first set of packets received from the first node including all packets containing information pertaining to the first source flow end point and all packets containing information pertaining to the second destination flow end point and performing the following four steps (a), (b), (c) and (d) on each packet in this first set of packets;

(a) processing any information in the packet pertaining to the second flow according to the second protocol;

(b) replacing any information in the packet pertaining to the second flow with the corresponding information pertaining to the fourth flow;

(c) modifying the packet so that any information in the packet pertaining to the flow from the first source flow end point will appear to the second node as pertaining to the flow to the third destination flow end point thus establishing a correspondence between data received by the intermediate node from the first source flow end point and data sent by the

intermediate node to the third destination flow end point, and

(d) sending each packet so processed to the second node;

identifying a second set of packets received from the second node including all packets containing information pertaining to the third destination flow end point and all packets containing information pertaining to the fourth source flow end point and performing the following steps (d), (e), and (f), on each packet in this second set of packets;

(d) identifying any information in the packet pertaining to the fourth flow and processing such information according to the fourth protocol;

(e) modifying any information in the packet pertaining to the flow to the third destination flow end point so the information instead pertains to the flow from the first source flow end point according to the correspondence between data received by the intermediate node from the first source flow end point and data sent by the intermediate node to the third destination flow end point established in step (b);

(f) sending to the first node zero or more packets containing any information resulting from step (e).

11. A method recited in claim 10, wherein step (c) includes determining if the packet to be sent should be fragmented and fragmenting it if so.

12. A method recited in claim 10, further comprising determining if each packet in the first set of packets contains data written to the first source flow end point and nullifying steps (b), (c) and (d) if not.

13. A method recited in claim 10, further comprising determining if each packet in the second set of packets contains data written to the fourth source flow end point, determining if each packet in the second set of packets contains no information pertaining to the flow to the third destination flow end point that has not already been processed by a previous application of step (e), and nullifying steps (e) and (f) for a given packet if both determinations are true for the given packet.

14. A method recited in claim 1, wherein the first protocol and the third protocol each associate a sequence number with each byte, packet or other unit of data sent across a flow further comprising the step of maintaining a one to one mapping between sequence numbers associated by the first protocol with each byte, packet, or other unit of data received by the intermediate node from the first source flow end point and sequence numbers associated by the second protocol with each byte, packet, or other unit of data sent by the intermediate node to the third destination flow end point.

15. A method recited in claim 1, further comprising the step of sending data stored in buffers associated with the third source flow end point to the third destination flow end point according to the third protocol.

16. A method recited in claim 1, further comprising the step of sending data stored in buffers associated with the first destination flow end point to the third destination flow end point according to the third protocol.

17. A method recited in claim 1, further comprising the step of determining if the first node has shut down the flow originating at the first source flow end point and shutting down the flow terminating at the third destination flow end point if this determination is true.

18. A method recited in claim 1, further comprising determining if the first node has shut down the flow originating at the first source flow end point and eliminating the first composite flow and recreating the third flow if this determination is true.

19. A method recited in claim 1, further comprising the step of making a copy of data received from the first node by the intermediate node such that the copy may subsequently be read at the intermediate node.

20. A method recited in claim 1, further comprising the steps of:

reading an amount of data from the fourth destination flow end point;

storing the amount of data in a format satisfying a need from the group of needs consisting of:

reducing the number of bits required to store the data,

encrypting the data, reducing the number or amount of resources required to transmit the data,

reducing the number or amount of resources required to display the data, and

any combination of these needs; and

sending the stored data to the first node.

21. An apparatus comprising:

a first node, a second node, an intermediate node;

a first processor that forms a first connection between a first node and the intermediate node the first connection including:

a first flow originating at a first source flow end point on the first node and terminating at a first destination flow end point on the intermediate node, wherein processing at the first node associated with the first source flow end point conforms to a first protocol, and

a second flow originating at a second source flow end point on the intermediate node and terminating at a second destination flow end point on the first node, wherein processing at the first node associated with the second destination flow end point conforms to a second protocol;

a second processor that forms a second connection between a second node and the intermediate node the second connection including:

a third flow originating at a third source flow end point on the intermediate node and terminating at a third destination flow end point on the second node, wherein processing at the second node associated with the third destination flow end point conforms to a third protocol, and

a fourth flow originating at a fourth source flow end point on the second node and terminating at a fourth destination flow end point on the intermediate node, wherein processing at the second node associated with the fourth source flow end point conforms to a fourth protocol,

such that a given flow originates at a source flow end point on a source node and terminates at a destination flow end point on a destination node and data written to the source flow end point of a given flow can subsequently be read from the destination

flow end point of the given flow without traversing any intervening flow end points, and

a third processor that splices the first flow and third flow to form a first composite flow originating at the first source flow end point on the first node and terminating at the third destination flow end point on the second node in a manner whereby the second flow and the fourth flow remain unchanged.

22. A method recited in claim 1, further comprising:

forming a third connection between a third node and the intermediate node the third connection including:

a fifth flow originating at a fifth source flow end point on the intermediate node and terminating at a fifth destination flow end point on the third node, wherein processing at the third node associated with the fifth destination flow end point conforms to a fifth protocol, and

a sixth flow originating at a sixth source flow end point on the third node and terminating at a sixth destination flow end point on the intermediate node, wherein processing at the sixth node associated with the fourth source flow end point conforms to a sixth protocol,

splicing the sixth flow and second flow to form a third composite flow originating at the sixth source flow end point on the third node and terminating at the

second destination flow end point on the first node in a manner whereby the fourth flow and the fifth flow remain unchanged.

23. A method recited in claim 1, further comprising the step of reading data from the first destination flow end point prior to the step of splicing.

24. A method recited in claim 1, further comprising the step of writing data to the third source flow end point prior to the step of splicing.

25. A method recited in claim 1, further comprising the step of determining if the step of splicing should be performed based on a criterion selected from the group of criteria consisting of:

data read from the first destination flow end point,

data read from the fourth destination flow end point,

data written to the second source flow end point,

data written to the third source flow end point,

an address associated with the first node,

an address associated with the second node,

an address associated with the intermediate node,

an address associated with the first flow,

an address associated with the second flow,

an address associated with the third flow,

an address associated with the fourth flow,

an estimate of the available bandwidth along the first flow,

an estimate of the available bandwidth along the second flow,

an estimate of the available bandwidth along the third flow,
an estimate of the available bandwidth along the fourth flow,
an estimate of the end to end latency associated with the first flow,
an estimate of the end to end latency associated with the second flow,
an estimate of the end to end latency associated with the third flow,
an estimate of the end to end latency associated with the fourth flow,
a receive window size advertised by the first node,
a receive window size advertised by the second node,
a receive window size advertised by the intermediate node,
a number of network hops between the first node and the intermediate node,
a number of network hops between the intermediate node and the second node,
a number of network hops between the first node and the second node,
a protocol associated with the first flow,
a protocol associated with the second flow,
a protocol associated with the third flow,
a protocol associated with the fourth flow,
an estimate of the amount of data that will be written to the first source flow end point,
an estimate of the amount of data that will be written to the second source flow end point,
an estimate of the amount of data that will be written to the third source flow end point,

an estimate of the amount of data that will be written to the fourth source flow end point,
configuration information stored at or accessible from the intermediate node,
information received from the first node pertaining to quality of service,
information received from the second node pertaining to quality of service , and
any combination of these criteria.

26. A method recited in claim 1, further comprising the step of determining the network addresses with which the first flow, second flow, third flow, and fourth flow are established based on a criterion selected from the group of criteria consisting of:
data read from the first destination flow end point,
data read from the fourth destination flow end point,
data written to the second source flow end point,
data written to the third source flow end point,
a network address associated with a potential or actual first node,
a network address associated with a potential or actual second node,
a network address associated with the intermediate node,
an address associated with the first flow,
an address associated with the second flow,
an address associated with the third flow,
an address associated with the fourth flow,
an estimate or prediction of the available bandwidth along the first flow

an estimate or prediction of the available bandwidth along the second flow,
an estimate or prediction of the available bandwidth along the third flow
an estimate or prediction of the available bandwidth along the fourth flow,
an estimate or prediction of the end to end latency along the first flow,
an estimate or prediction of the end to end latency along the second flow,
an estimate or prediction of the end to end latency along the third flow,
an estimate or predication of the end to end latency along the fourth flow,
a receive window size advertised by the first node,
a receive window size advertised by the second node,
a receive window size advertised by the intermediate node,
a number of network hops between the first node and the intermediate node,
a number of network hops between the intermediate node and the second node,
a number of hops between the first node and the second node,
a protocol associated with the first flow,
A protocol associated with the second flow,
an estimate of the amount of data that will be written to the first source flow end point,
an estimate of the amount of data that will be written to the second source flow end point,
an estimate of the amount of data that will be written to the third source flow end point,

an estimate of the amount of data that will be written to the fourth source flow end point,

configuration information stored at or accessible from the intermediate node,

information received from the first node pertaining to quality of service,

information received from the second node pertaining to quality of service, and

any combination of these criteria.

27. A method recited in claim 10, wherein each of the first protocol, the second protocol, the third protocol and the fourth protocol are Transmission Control Protocol (TCP) in conjunction with either version 4 or version 6 of Internet Protocol (IP) and wherein step (c) of claim 1, includes the following steps (a), (b), (c), (d), and (e):

(a) setting the source IP address in the IP header to the local IP address associated with the third source flow end point;

(b) setting the destination IP address in the IP header to the remote IP address associated with the third source flow end point;

(c) setting the source port number in the TCP header to the local port number associated with the third source flow end point;

(d) setting the destination port number in the TCP header to the remote port number associated with the third source flow end point, and

(e) modifying the sequence (SEQ) number in the TCP header;

and wherein step (b) of claim 10, includes the following steps (f), (g), (h), and (i):

(f) replacing the acknowledgment (ACK) number in the TCP header;

(g) replacing the window value in the TCP header;

(h) modifying or recalculating the TCP checksum in the TCP header, and

(i) modifying or recalculating the IP checksum in the IP header;

and wherein step (e) of claim 1, includes the following steps (j), (k), (l), (m), (n), (o), (p) and (q):

(j) setting the source IP address in the IP header to the local IP address associated with the first destination flow end point;

(k) setting the destination IP address in the IP header to the remote IP address associated with the first destination flow end point;

(l) setting the source port number in the TCP header to the local port number associated with the first destination flow end point;

(m) setting the destination port number in the TCP header to the remote port number associated with the first destination flow end point;

(n) replacing the sequence (SEQ) number in the TCP header;

(o) modifying the acknowledgment (ACK) number in the TCP header;

(p) modifying the TCP checksum in the TCP header, and

(q) modifying the IP checksum in the IP header.

28. A method recited in claim 27, wherein any of the checksum calculations performed in steps (h), (i), (p), and (q) is calculated by adjusting the original checksum to account for changes made to the packet at the intermediate node.

29. A method recited in claim 27, wherein the first connection extends between a TCP server on the first node and a SOCKS server on the intermediate node and the second connection extends between the SOCKS server on the intermediate node and a SOCKS client on the second node further comprising parsing data received from either the SOCKS client or the TCP server.

30. A method recited in claim 27, wherein the first connection extends between a SOCKS client on the first node and a SOCKS server on the intermediate node and the second connection extends between the SOCKS server on the intermediate node and a TCP server on the second node further comprising parsing data received from either the SOCKS client or the TCP server.

31. A method recited in claim 27, wherein the first connection extends between a Sun remote procedure call (Sun RPC) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and a Sun RPC client on the second node further comprising

parsing data received from either the Sun RPC client or the Sun RPC server.

32. A method recited in claim 27, wherein the first connection extends between a Sun remote procedure call (Sun RPC) client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and a Sun RPC server on the second node further comprising parsing data received from either the Sun RPC client or the Sun RPC server.

33. A method recited in claim 27, wherein the first connection extends between a hypertext transfer protocol (HTTP) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an HTTP client on the second node further comprising parsing data received from either the HTTP client or the HTTP server.

34. A method recited in claim 27, wherein the first connection extends between a hypertext transfer protocol (HTTP) client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an HTTP server on the second node further comprising parsing data received from either the HTTP client or the HTTP server.

35. A method recited in claim 27, wherein the first connection extends between a file transfer protocol

(FTP) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an FTP client on the second node further comprising parsing data received from either the FTP client or the FTP server.

36. A method recited in claim 27, wherein the first connection extends between a file transfer protocol (FTP) client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an FTP server on the second node further comprising parsing data received from either the FTP client or the FTP server.

37. A method recited in claim 27, wherein the first connection extends between a telnet server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an telnet client on the second node further comprising parsing data received from either the telnet client or the telnet server.

38. A method recited in claim 27, wherein the first connection extends between a telnet client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an telnet server on the second node further comprising parsing data received from either the telnet client or the telnet server.

39. A method recited in claim 27, wherein the first connection extends between a network news transfer protocol (NNTP) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an NNTP client on the second node further comprising parsing data received from either the NNTP client or the NNTP server.

40. A method recited in claim 27, wherein the first connection extends between a network news transfer protocol (NNTP) client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an NNTP server on the second node further comprising parsing data received from either the NNTP client or the NNTP server.

41. A method recited in claim 27, wherein the first connection extends between a secure shell (SSH) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an SSH client on the second node further comprising parsing data received from either the SSH client or the SSH server.

42. A method recited in claim 27, wherein the first connection extends between a secure shell (SSH) client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an SSH server on the second

node further comprising parsing data received from either the SSH client or the SSH server.

43. A method recited in claim 27, wherein the first connection extends between a remote shell (RSH) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an RSH client on the second node further comprising parsing data received from either the RSH client or the RSH server.

44. A method recited in claim 27, wherein the first connection extends between a remote shell (RSH) client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an RSH server on the second node further comprising parsing data received from either the RSH client or the RSH server.

45. A method recited in claim 27, wherein the first connection extends between a simple mail transfer protocol (SMTP) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and an SMTP client on the second node further comprising parsing data received from either the SMTP client or the SMTP server.

46. A method recited in claim 27, wherein the first connection extends between a simple mail transfer protocol (SMTP) client on the first node and a proxy on the intermediate node and the second connection extends

between the proxy on the intermediate node and an SMTP server on the second node further comprising parsing data received from either the SMTP client or the SMTP server.

47. A method recited in claim 27, wherein the first connection extends between a post office protocol (POP) server on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and a POP client on the second node further comprising parsing data received from either the POP client or the POP server.

48. A method recited in claim 27, wherein the first connection extends between a post office protocol (POP) client on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and a POP server on the second node further comprising parsing data received from either the POP client or the POP server.

49. A method recited in claim 27, wherein the first connection extends between a server sending streaming audio and or video on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and a client receiving streaming audio or video on the second node further comprising parsing data received from either the client or the server.

50. A method recited in claim 37 wherein the first connection extends between a client reading streaming audio and/or video on the first node and a proxy on the intermediate node and the second connection extends between the proxy on the intermediate node and a server sending streaming audio or video on the second node further comprising parsing data received from either the client or the server.

51. A method recited in claim 1, further comprising:

forming a third connection between a third node and the intermediate node the third connection including:

a fifth flow originating at a fifth source flow end point on the intermediate node and terminating at a fifth destination flow end point on the third node, wherein processing at the third node associated with the fifth destination flow end point conforms to a fifth protocol, and

a sixth flow originating at a sixth source flow end point on the third node and terminating at a sixth destination flow end point on the intermediate node, wherein processing at the sixth node associated with the fourth source flow end point conforms to a sixth protocol,

splicing the fourth flow and fifth flow to form a second composite flow originating at the fourth source flow end point on the second node and terminating at the fifth destination flow end point on the third node

in a manner whereby the second flow and the sixth flow remain unchanged.

52. An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for causing the processing of network packets at an intermediate node, the computer readable program code means in said article of manufacture comprising computer readable program code means for causing a computer to effect:

forming a first connection between a first node and the intermediate node this first connection including:

a first flow originating at a first source flow end point on the first node and terminating at a first destination flow end point on the intermediate node, wherein processing at the first node associated with the first source flow end point conforms to a first protocol, and

a second flow originating at a second source flow end point on the intermediate node and terminating at a second destination flow end point on the first node, wherein processing at the first node associated with the second destination flow end point conforms to a second protocol;

forming a second connection between a second node and the intermediate node this second connection including:

a third flow originating at a third source flow end point on the intermediate node and terminating at a third destination flow end point on the second node, wherein processing at the second node associated with the third destination flow end point conforms to a third protocol, and

a fourth flow originating at a fourth source flow end point on the second node and terminating at a fourth destination flow end point on the intermediate node, wherein processing at the second node associated with the fourth source flow end point conforms to a fourth protocol,

such that a given flow originates at a source flow end point on a source node and terminates at a destination flow end point on a destination node and data written to the source flow end point of a given flow can subsequently be read from the destination flow end point of the given flow without traversing any intervening flow end points, and

splicing the first flow and third flow to form a first composite flow originating at the first source flow end point on the first node and terminating at the third destination flow end point on the second node in a manner whereby the second flow and the fourth flow remain unchanged.

53. An article of manufacture as recited in claim 52, wherein the step of splicing the first flow and third flow to form the first composite flow allows:

processing at the first node associated with the first source flow end point to remain unmodified and continue to conform to the first protocol,

processing at the first node associated with the second destination flow end point to remain unmodified and continue to conform to the second protocol,

processing at the second node associated with the third destination flow end point to remain unmodified and continue to conform to the third protocol, and

processing at the second node associated with the fourth source flow end point to remain unmodified and continue to conform to the fourth protocol.

54. An article of manufacture as recited in claim 52, wherein the first node and the second node are the same node.

55. An article of manufacture as recited in claim 52, wherein the first protocol and the third protocol each associate a sequence number with each byte, packet or other unit of data sent across a flow, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect maintaining a one to one mapping between sequence numbers associated by the first protocol with each byte, packet, or other unit of data received by the intermediate node from the first source flow end point and sequence numbers associated by the second protocol with each byte, packet, or other unit of data sent by the intermediate node to the third destination flow end point.

56. An article of manufacture as recited in claim 52, wherein the step of splicing the first flow and third flow to form the first composite flow includes:

identifying a first set of packets received from the first node including all packets containing information pertaining to the first source flow end point and all packets containing information pertaining to the second destination flow end point and performing the following four steps (a), (b), (c) and (d) on each packet in this first set of packets;

(a) processing any information in the packet pertaining to the second flow according to the second protocol;

(b) replacing any information in the packet pertaining to the second flow with the corresponding information pertaining to the fourth flow;

(c) modifying the packet so that any information in the packet pertaining to the flow from the first source flow end point will appear to the second node as pertaining to the flow to the third destination flow end point thus establishing a correspondence between data received by the intermediate node from the first source flow end point and data sent by the intermediate node to the third destination flow end point, and

(d) sending each packet so processed to the second node;

identifying a second set of packets received from the second node including all packets containing information pertaining to the third destination flow end point and all packets containing information pertaining to the fourth source flow end point and performing the following steps (d), (e), and (f), on each packet in this second set of packets;

(d) identifying any information in the packet pertaining to the fourth flow and processing such information according to the fourth protocol;

(e) modifying any information in the packet pertaining to the flow to the third destination flow end point so the information instead pertains to the flow from the first source flow end point according to the correspondence between data received by the intermediate node from the first source flow end point and data sent by the intermediate node to the third destination flow end point established in step (b);

(f) sending to the first node zero or more packets containing any information resulting from step (e).

57. An article of manufacture as recited in claim 56, wherein each of the first protocol, the second protocol, the third protocol and the fourth protocol are Transmission Control Protocol (TCP) in conjunction with either version 4 or version 6 of Internet Protocol

(IP) and wherein step (c) of claim 1, includes the following steps (a), (b), (c), (d), and (e):

(a) setting the source IP address in the IP header to the local IP address associated with the third source flow end point;
(b) setting the destination IP address in the IP header to the remote IP address associated with the third source flow end point;
(c) setting the source port number in the TCP header to the local port number associated with the third source flow end point;
(d) setting the destination port number in the TCP header to the remote port number associated with the third source flow end point, and
(e) modifying the sequence (SEQ) number in the TCP header;

and wherein step (b) of claim 10, includes the following steps (f), (g), (h), and (i):

(f) replacing the acknowledgment (ACK) number in the TCP header;
(g) replacing the window value in the TCP header;
(h) modifying or recalculating the TCP checksum in the TCP header, and
(i) modifying or recalculating the IP checksum in the IP header;

and wherein step (e) of claim 1, includes the following steps (j), (k), (l), (m), (n), (o), (p) and (q):

(j) setting the source IP address in the IP header to the local IP address associated with the first destination flow end point;

(k) setting the destination IP address in the IP header to the remote IP address associated with the first destination flow end point;

(l) setting the source port number in the TCP header to the local port number associated with the first destination flow end point;

(m) setting the destination port number in the TCP header to the remote port number associated with the first destination flow end point;

(n) replacing the sequence (SEQ) number in the TCP header;

(o) modifying the acknowledgment (ACK) number in the TCP header;

(p) modifying the TCP checksum in the TCP header, and

(q) modifying the IP checksum in the IP header.

58. A computer program product comprising a computer usable medium having computer readable program code means embodied therein for causing the processing of network packets at an intermediate node, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect:

forming a first connection between a first node and the intermediate node this first connection including:

a first flow originating at a first source flow end point on the first node and terminating at a first destination flow end point on the intermediate node, wherein processing at the first node associated with the first source flow end point conforms to a first protocol, and

a second flow originating at a second source flow end point on the intermediate node and terminating at a second destination flow end point on the first node, wherein processing at the first node associated with the second destination flow end point conforms to a second protocol;

forming a second connection between a second node and the intermediate node this second connection including:

a third flow originating at a third source flow end point on the intermediate node and terminating at a third destination flow end point on the second node, wherein processing at the second node associated with the third destination flow end point conforms to a third protocol, and

a fourth flow originating at a fourth source flow end point on the second node and terminating at a fourth destination flow end point on the intermediate node, wherein processing at the second node associated with the fourth source flow end point conforms to a fourth protocol,

such that a given flow originates at a source flow end point on a source node and terminates at a destination flow end point on a destination node and data written to the source flow end point of a given flow can subsequently be read from the destination flow end point of the given flow without traversing any intervening flow end points, and

splicing the first flow and third flow to form a first composite flow originating at the first source flow end point on the first node and terminating at the third destination flow end point on the second node in a manner whereby the second flow and the fourth flow remain unchanged.